Target-matrix Construction Algorithms

Patrick Knupp*
Applied Mathematics & Applications Department,
Sandia National Laboratories,
M/S 1318, P.O. Box 5800,
Albuquerque, NM 87185-1318
(pknupp@sandia.gov)

30 September 2009

Abstract

A gap in the series of reports describing the Target-matrix paradigm for mesh optimization is filled by describing a method for automatic construction of Target-matrices, so that diverse applications such as shape improvement, geometric-adaptivity, solution-adaptivity, mesh alignment, and anisotropic smoothing can be served while using only a limited set of local quality metrics. The method makes use of the Jacobian matrices at sample points in the initial mesh. A QR-factorization is applied to the Jacobian matrix to isolate different initial mesh properties such as Size, Shape, and Orientation. To construct a Target-matrix, each factor can be retained if the goal is to preserve the property of the initial mesh, or it can be replaced by a new factor based on application or other data if the goal is to improve the initial property. By this method, one can rapidly implement custom-made mesh optimization algorithms in response to requests from application groups desiring improved meshes in order to perform more accurate and efficient simulations. An example from a Cubit application is given.

1 Introduction

This work is part of a series of papers [1]-[5] which are devoted to a description and analysis of the Target-matrix paradigm (TMP) for mesh optimization. Previous papers in the series have focused on the overall TMP formulation, objective functions, and investigations into local quality metrics (including barriers, convexity, derivatives, and other essential properties). The present paper fills a critical gap, namely a description of methods for automatically constructing the Target-matrices. Target-matrices are required in all TMP optimization problems.

Targets play a critical role in the paradigm because they are the means by which applications describe the particular mesh optimization problem they wish to solve. There are, perhaps, a dozen distinct high-level applications of mesh optimization including shape improvement, geometric-adaptivity, solution-adaptivity, mesh alignment, and anisotropic smoothing. In TMP, each application requires a different set of Target-matrices that, along with the local quality metric, precisely define the goal of the optimization. For example, to adapt a mesh to the local discretization error requires a different set of target-matrices than the problem of optimizing a mesh to increase minimum edge-length. TMP moves the burden in mesh optimization from designing or selecting good local quality metrics to the automatic

*This document is SAND 2009-XXX. This work was funded by the Department of Energy's Mathematics, Information, and Computational Science Program (SC-31) and was performed at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy under contract DE-ACO4-94AL85000. This work was performed by an employee of the U.S. Government or under U.S. Government contract.

construction of target matrices derived from application goals. As with other methods, construction of targets remains somewhat of an art, but in TMP is made more tractable for two reasons. First, the targets are based on the Jacobian matrix of the desired optimal mesh and thus have a simple geometric interpretation. Second, the target construction method makes use of the initial mesh (the one to be optimized). The initial mesh is nearly always available, and, in most other optimization methods, is ignored, even though it often contains valuable information. Before describing methods for target construction, a brief review of the TMP follows.

First, one defines a set of local mappings from points Ξ in the master element(s) to points $X(\Xi)$ in each of the elements in the mesh that is to be optimized (the latter is called the *active* mesh). These mappings are most commonly those from linear finite elements, but can be more general if needed. If the active mesh consists of only one element type, then the mappings can all have the same form, e.g., the linear map from a square to a quadrilateral. If the active mesh contains more than one element type (e.g., tetradehra and triangular wedges) then more than one mapping form is required. Although the form of the mapping may be the same from one element to another, the exact mapping on each element can differ because the mapping depends on the coordinates of the vertices which define the particular element. Non-linear mappings are also allowed, for example, in the case of high-order finite elements.

In addition to the mappings, TMP requires that a set of *sample points* within the master element(s) be selected. Let the sample points within the master element be denoted by $\{\Xi_k\}$, $k = 0, 1, \ldots, K-1$. The corresponding points in the active mesh are $\{X_k\}$ where $X_k = X(\Xi_k)$. Typically, the sample points are located at the corners of the master element if the element is linear, otherwise they may also be located at mid-edges, mid-faces, and/or mid-elements. TMP thus requires that, in the formulation stage of the optimization, one define a set of mappings and sample points over all the elements of the mesh. This is not as daunting as it sounds because the mappings are usually of the same form for each element in the mesh (unless it contains more than one element type) and thus there is only one master element that is used for every element in the active mesh. The sample points are usually located at the corners of the master element if the mapping is linear.

The mappings are required to be differentiable so that their Jacobian matrix $\partial X/\partial \Xi$ exists at the sample points. For short-hand, we denote this Jacobian matrix by the symbol A, which refers to the Jacobian of the map from the master element to an element in the *active* mesh. The Jacobian matrix depends upon the vertex coordinates within each element and thus varies from one element to the next; this dependence is suppressed in the notation used above. Furthermore, the Jacobian matrix varies from point to point within the master element, as a function of Ξ. We denote by A_k the Jacobian matrix evaluated at sample point k; thus $A_k = A(\Xi_k)$.

There is yet another requirement in TMP, namely the creation of a set of Target-matrices; the requirement is more difficult to achieve, but gives TMP considerable power and flexibility. For every sample point k in the mesh, the target paradigm requires two matrices: the Jacobian matrix A_k derived from the active mesh and the Target (or reference-Jacobian)-matrix W_k. The set matrices $\{A_k\}$ is readily computed from the active mesh, while the set $\{W_k\}$ must be constructed prior to the optimization. This construction is the subject of the present paper. The purpose of the Target-matrices is to quantitatively describe the optimal mesh Jacobians on the set of sample points. Since every sample point can have a different Target-matrix, this information is very detailed and difficult to determine from scratch. Fortunately, there are mitigating factors (to be described later) which can be taken advantage of to find suitable target sets.

For clarity, the sample point indices are often suppressed in much of the remainder of this presentation. Let A and W at some sample point be defined. Because the construction of targets is under our control, we can assume that, for every target, $det(W) \neq 0$ and thus W^{-1}

exists. The *weighted Jacobian* matrix T, defined by $T = AW^{-1}$, is heavily used in TMP because, if the target matrix W has units of length (as does A), then T is non-dimensional and provides a convenient scaling of A.

Before getting into the details of how the Targets are created, let us complete the description of TMP. Let M_d be the set of $d \times d$ matrices with real numbers as elements. In mesh optimization the matrices A, W, and T are either 2×2 or 3×3, reflecting the dimension of the elements in the mesh.[1]. A local quality metric μ, which is a function from M_d to the non-negative numbers, is chosen from a set of quality metrics given in previous TMP papers; for example, the Size+Shape+Orientation metric $\mu(T) = |T - I|^2$ is frequently used, as are Shape and Size+Shape metrics. Note that both A and T depend on the coordinates of the vertices within each element. An objective function, typically of the form

$$F = \frac{1}{N} \sum_e \sum_k \mu(T_k^e)$$

where k is the sample point index within an element, e is the element index, and N is the total number of sample points in the mesh, is minimized as a function of the coordinates of the free vertices in the mesh to find the optimal mesh. The optimization is usually constrained by fixing some or all of the boundary vertices.

The above optimization problem, applied to most meshes, cannot usually be solved without resorting to iterative numerical methods. For the most part, standard optimization methods are used to solve the TMP optimization problem. Iterative methods require that we begin with an *initial* mesh. Once the mappings and sample points in TMP are defined, we can readily compute the Jacobian matrix A_{init} on the initial mesh. This matrix plays an important role, not only in the initialization of the optimization problem, but (as we shall see) in the construction of the target-matrices.

2 Two Simple Ways to Construct the Target-Matrices

So far, we have mentioned both the master and active mesh elements and the Jacobian matrix A which relates the two. If one were, in addition, to define a *reference* element which specifies the target (or optimal) element in the mesh, then the picture in Figure 1 shows the relation between three different maps and their corresponding Jacobian matrices A, W, and T. The target matrix is thus the Jacobian of the map between the master and reference elements, while the weighted Jacobian matrix is the Jacobian of the map between the reference element and the physical element. The three maps have the same mathematical form (e.g., the bilinear map for quadrilateral elements), but of course will differ otherwise because the vertex coordinates of the logical, reference, and physical elements are not generally the same.

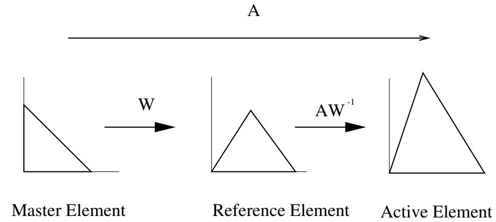

Figure 1: Relation between the Master, Reference, and Active Elements

[1]The mesh is assumed to be confined to either R^2 or to R^3, i.e., either a planar or a volume mesh; the surface case is reserved for a later paper

The figure suggests two relatively simple methods for constructing the set of target matrices. In the first method, the reference element is held fixed over all the elements in the mesh. In this case, the reference element represents the ideal element for a given element-type. Thus, for example, if the mesh to be optimized consists of triangular elements, then the reference element can be taken to be an equilateral triangle, whereas if the mesh consists of quadrilaterals, then the reference element can be the square. These reference elements are useful if the goal of the optimization is to create an optimal mesh consisting of all-equilateral elements. Given the ideal reference element, one can determine the target W simply by evaluating the Jacobian of the map from the master to the reference element at each of the sample points (see [6] for example). In this approach, the ideal targets are used in conjunction with a Shape metric such as condition number, which is size and orientation invariant, so that the equilateral reference element can have arbitrary size and orientation.[2] Clearly, this particular method of constructing target-matrices, while simple, effective, and sometimes appropriate for the application, can only create meshes whose element shapes are close to the ideal shape. Other methods of target construction are needed if the goal of the optimization is different.

In the second method of target construction, which is referred to as the *reference mesh* method, the reference element in the figure becomes an element from a reference mesh that is topologically-identical to the active mesh. Given the reference mesh, the sample points, and the same element mapping type(s) used to compute A from the active mesh, let the Jacobian matrices of the reference mesh be A_{ref}; this is readily computed using the vertex coordinates of the reference mesh. The target is then taken to be $W = A_{ref}$. Choosing the target matrix in this manner implies that the reference mesh has good quality and is thus suitable as the target of the optimization. The challenge in this method, of course, is to obtain a suitable reference mesh. An example is found in the 'deforming geometry' problem in which one is asked to update the mesh as the domain on which it is defined changes shape (often vs. time). In this problem, the mesh on the geometry before it is deformed is used as the reference mesh and the metric $|T - I|^2$ may be suitable. At a particular time t during the deformation $T = A(t)(A_{ref})^{-1}$. The optimization will drive the metric towards zero so that $T \approx I$ and so $A(t) \approx A_{ref}$. In that case, the optimal mesh consists of elements that are close (in a least-squares sense) to the reference mesh. Details of this method are given in [7].

Another example of using a reference mesh occurs in the mesh copy/morph problem in which one seeks to improve a mesh on one geometry when given a good quality mesh on a topologically similar geometry. For example, many geometries in engineering are quasi-cylindrical. Given an imperfect mesh on such a geometry, one may be able to improve it using the reference mesh method. In this approach, one first creates a topologically-identical mesh of good quality using a exactly cylindrical geometry (perhaps by using the cylindrical coordinates transformation or by using a sweeping algorithm). This latter mesh is then used as the reference mesh in optimizing the mesh on the quasi-cylindrical geometry.

A special case of the reference mesh method results when one chooses the *initial* mesh (which is used to begin the iterative solution method) as the reference mesh. Then $W = A_{init}$ and $T = A(A_{init})^{-1}$. Thus, at the beginning of the optimization $T_{init} = A_{init}(A_{init})^{-1} = I$. But $T = I$ is a global minimizer of most of the TMP quality metrics, so the optimal mesh is simply the initial mesh. In most cases this is not useful, although there are some exceptions (see [8], for example).

The two simple methods described above for constructing target matrices enable optimization procedures for creating meshes with all elements close to an ideal element or to create a good quality mesh on one geometry given a good quality (topologically-identical) mesh on a

[2] It is important to understand that to create a mesh with equilateral elements, one not only has to create the proper reference element, but also must use the proper local quality metric. For example, if one used the equilateral reference element with the Size+Shape+Orientation metric, the optimal mesh would have local orientations all the same. The result would most likely be an inverted mesh.

similar geometry. To meet additional mesh optimization goals requires a more sophisticated approach to target construction. In particular, we shall move away from the requirement of having to provide a set of reference elements like those in Figure 1. Instead, the targets will be constructed directly from application data, including the initial mesh. As before, one Target-matrix will be constructed for each sample point within the mesh. This set of matrices will not necessarily be self-consistent. For example, it only takes two consistent Jacobian matrices to uniquely determine (up to translation) the vertex coordinates of a linear quadrilateral. If we create four independent Jacobian matrices, the quadrilateral reference element may not actually exist. However, consistency is not required because the least-squares formulation of the objective function will reconcile any inconsistencies which exist between the targets within elements and between different elements.[3]

As we shall see, the initial mesh can be useful in constructing target matrices if the quality of the initial mesh is already partly acceptable.[4] For example, perhaps the size of the elements in the initial mesh is acceptable while the shape leaves something to be desired. We would thus want to construct target matrices whose size corresponds to the sizes in the initial mesh, but whose shape is given by the ideal element shape. Somewhat surprisingly, this can in fact be accomplished using a certain matrix factorization which we now describe.

3 The $\Lambda\, V\, Q\, \Delta$ Factorization

The matrix factorization is related to the well-known 'QR-decomposition' in which a matrix is factored as the product of an orthogonal and an upper triangular matrix. When the matrix that is factored is the 2×2 or 3×3 Jacobian matrix A, the factorization can be given explicitly, and the factors have identifiable geometric meanings. In this section attention is restricted to the 2×2 case in order to convey the basic results. Section 5 considers the 3×3 case, while the 3×2 surface case is deferred until a later work.

Let M_2 be the set of all 2×2 matrices with real elements. Also define M_2^* to be the set of all matrices in M_2 that are non-singular. Consider $A_{2\times 2}$, with the columns of A given by the vectors a_1 and a_2, and write $A = [a_1, a_2]$. If $A \in M_2^*$, then $\alpha \equiv det(A) \neq 0$ and, further, the lengths $|a_1|$ and $|a_2|$ of the two column vectors are non-zero. When A is the Jacobian matrix, the two column vectors correspond to the two tangent vectors of the mapping (at a given sample point), $a_1 \cdot a_2$ to the angle between the vectors, and $det(A)$ to the area of the parallelogram defined by the vectors.

The factorization of interest requires that $A \in M_2^*$. In the theory to follow we shall assume that this is the case.[5] There are actually three factorizations of interest. The factorizations of $A \in M_2^*$ that we are interested in are given explicitly by

$$
\begin{aligned}
A &= V\, U && (1)\\
&= \Lambda\, V\, S && (2)\\
&= \Lambda\, V\, Q\, \Delta && (3)
\end{aligned}
$$

where

$$\Lambda(A) = \sqrt{|\alpha|} \qquad (4)$$

[3]This is true for the previously-described ideal reference element case, too, since consistency of Jacobians between ideal elements is not guaranteed.

[4]Even if it is not, the initial mesh can provide a rough idea of the local Sizes within the mesh. This is important because the Size+Shape metric, for example, requires exact sizes, not relative sizes.

[5]In practice, to deal with singular Jacobian matrices arising from the initial mesh we shall take the pragmatic approach of judiciously applying a sufficiently small random perturbation to one or more vertices within the element that gave rise to the singular matrix. The result almost always will be a non-singular matrix that is close to the original singular matrix but to which the factorization applies.

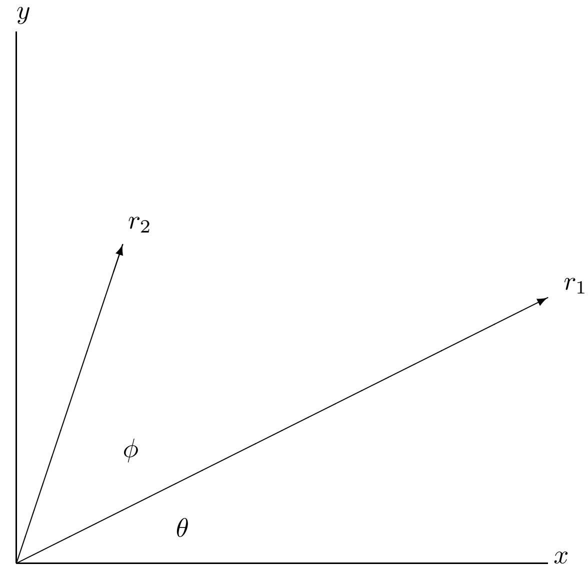

Figure 2: The vectors $a_1 = r_1\,(\cos\theta, \sin\theta)$ and $a_2 = r_2\,(\cos(\theta+\phi), \sin(\theta+\phi))$.

is a scalar, and the remaining factors are the 2×2 matrices:

$$V(A) = \left[\frac{a_1}{|a_1|}, \frac{-(a_1 \cdot a_2)\,a_1 + |a_1|^2\,a_2}{|\alpha|\,|a_1|}\right] \quad (5)$$

$$Q(A) = \sqrt{\frac{|a_1|\,|a_2|}{|\alpha|}} \begin{pmatrix} 1 & \frac{a_1 \cdot a_2}{|a_1|\,|a_2|} \\ 0 & \frac{|\alpha|}{|a_1|\,|a_2|} \end{pmatrix} \quad (6)$$

$$\Delta(A) = \frac{1}{\sqrt{|a_1|\,|a_2|}} \begin{pmatrix} |a_1| & 0 \\ 0 & |a_2| \end{pmatrix} \quad (7)$$

$$S(A) = \frac{1}{\sqrt{|\alpha|}} \begin{pmatrix} |a_1| & \frac{a_1 \cdot a_2}{|a_1|} \\ 0 & \frac{|\alpha|}{|a_1|} \end{pmatrix} \quad (8)$$

$$U(A) = \begin{pmatrix} |a_1| & \frac{a_1 \cdot a_2}{|a_1|} \\ 0 & \frac{|\alpha|}{|a_1|} \end{pmatrix} \quad (9)$$

The explicit factorizations show that, given A non-singular, the 2×2 matrices V, Q, Δ, $S = Q\,\Delta$, and $U = \Lambda\,S$ are determined, as is the scalar Λ. This factorization of the mesh Jacobian was first mentioned in [9]. Note that the matrix factors themselves are non-singular.

The factors of A have recognizable geometric meanings. This is more easily seen if we replace the column vectors a_1 and a_2 in $A = [a_1, a_2]$ with the geometric quantities $r_1 = |a_1|$, $r_2 = |a_2|$, θ, and ϕ shown in Figure 2. Then ϕ is the angle between the two column vectors and θ is the angle between the first column vector and the x-axis. Let the Jacobian-matrix thus be given by

$$A = \begin{pmatrix} r_1 \cos\theta & r_2 \cos(\theta+\phi) \\ r_1 \sin\theta & r_2 \sin(\theta+\phi) \end{pmatrix}$$

Then $det(A) = r_1\,r_2\,\sin\phi$. The factorization thus exists provided $r_1 \neq 0$, $r_2 \neq 0$, $\phi \neq 0$, and $\phi \neq \pi$. Moreover, the conditions $r_1 > 0$, $r_2 > 0$, and $0 < \phi < \pi$ ensure that A has a positive determinant and thus V will be a rotation matrix, rather than a flip.

Applying the factorization formulas to the previous form of A, and assuming $det(A) > 0$, we obtain

$$\Lambda = \sqrt{r_1\,r_2\,\sin\phi}$$

$$
\begin{aligned}
V &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \\
Q &= \frac{1}{\sqrt{\sin\phi}} \begin{pmatrix} 1 & \cos\phi \\ 0 & \sin\phi \end{pmatrix} \\
\Delta &= \begin{pmatrix} \sqrt{\frac{r_1}{r_2}} & 0 \\ 0 & \sqrt{\frac{r_2}{r_1}} \end{pmatrix} \\
S &= \frac{1}{\sqrt{r_1\, r_2\, \sin\phi}} \begin{pmatrix} r_1 & r_2\, \cos\phi \\ 0 & r_2\, \sin\phi \end{pmatrix} \\
U &= \begin{pmatrix} r_1 & r_2\, \cos\phi \\ 0 & r_2\, \sin\phi \end{pmatrix}
\end{aligned}
$$

From this factorization, we see that Λ is the square root of the local area of the mapping. $\Lambda \geq 0$, is related to area and has units of length; we refer to Λ as the local *Size* parameter. The matrix V is dimensionless and is a rotation, i.e., $det(V) = 1$ and $V^t V = I$.[6] Because the first column of V is the unit vector in the direction a_1, we see that V is related to the local Orientation of the map with respect to the coordinate system. The matrix Q is dimensionless, upper triangular, has unit determinant, equal-length columns, and positive diagonal elements. Because Q can be written in terms of the sine and cosine of the angle between the vectors a_1 and a_2, we see that it is related to local angle or Skew between the tangents. The matrix Δ is dimensionless, diagonal, with positive diagonal elements, and unit determinant. Because the diagonal elements of Δ are ratios of the lengths of the two tangent vectors, we see that it is related to local aspect ratio. The matrix S is dimensionless, upper triangular, has positive diagonal entries, and unit determinant. As the product of skew and aspect ratio matrices, S is related to the local shape. The matrix U has units of length, is upper triangular, has positive diagonal entries, and its determinant is $|\alpha|$. As the product of Size and Shape quantities, we see that U is related to local Shape+Size.

To understand how this factorization can be used in target construction, recall the example given at the end of the previous section, in which the initial mesh had good Size quality, but poor Shape quality. The factorization shows that we can factor the Jacobian matrices belonging to the initial mesh into two matrices $\Lambda\, V$ and S such that $A_{init} = (\Lambda_{init}\, V_{init})\, S_{init}$. To construct an appropriate Target matrix, we discard S_{init} and replace it with S_{new}, the latter representing the desired local Shape. The Target-matrix is then $W = \Lambda_{init}\, V_{init}\, S_{new}$. A key question is: under what conditions is $S(W) = S_{new}$, i.e., how can one construct S_{new} such that the Shape implied by the Target is the shape specified by S_{new}?

To investigate this question, define the following matrix sets in M_2^*:

- $\mathcal{V}$ is the set of all orthogonal matrices, $\mathcal{R}$ is the set of all rotations, $\mathcal{F}$ is the set of all flips. The union of the latter two is $\mathcal{V}$,
- $\mathcal{U}$ is the set of all upper triangular matrices with positive diagonal elements,
- $\mathcal{S}$ is the set of all matrices in $\mathcal{U}$ having unit determinant,
- $\mathcal{Q}$ is the set of all matrices in $\mathcal{S}$ having equal length columns,
- $\mathcal{D}$ is the set of all diagonal matrices in $\mathcal{S}$

One can see by inspection of (4)-(9) that $V(A) \in \mathcal{V}$, $U(A) \in \mathcal{U}$, $S(A) \in \mathcal{S}$, $Q(A) \in \mathcal{Q}$, and $\Delta(A) \in \mathcal{D}$.

Proposition 1.
First, $det(A) > 0$ if and only if $V(A)$ is a rotation. Second, $det(A) < 0$ if and only if $V(A)$ is a flip.
Proof.

[6] Our choice of r_1, r_2, and ϕ eliminated the possibility that V could be a flip.

From (1) we see that $\alpha = det(A) = det(V)\, det(U) = det(V)\, |\alpha|$. Thus, if $det(A) > 0$, we must have $det(V) = 1$ and, because V is either a flip or a rotation, V must be a rotation. If, on the other hand, V is a rotation, then $det(V) = 1$, and so $det(A) = |\alpha| > 0$. The proof of the second statement is similar. §

Given $A \in M_2^*$, the matrices V, Q, Δ, etc. are uniquely determined. For the purpose of target construction, the key question is the reverse statement: given matrices V', U', under what conditions is $V(V'U') = V'$, $U(V'U') = U'$?

Proposition 2.
If $V' \in \mathcal{V}$, $U' \in \mathcal{U}$, and $A' = V'\, U'$, then $V(A') = V'$ and $U(A') = U'$.
Proof
The proof for $V' \in \mathcal{R}$ is provided here, while the case $V' \in \mathcal{F}$ is similar. Note first that $det(U') > 0$ since the determinant of an upper triangular matrix is the product of the diagonal entries and these are assumed positive. Since $det(V') = 1$, we have $det(A') = det(U') > 0$. Thus $A' \in M_2^*$ and the factorization $A' = V(A')\, U(A')$ exists. Therefore, we have

$$V'\, U' \quad = \quad V(A')\, U(A')$$

From this, $det(V')\, det(U') = det[V(A')]\, det[U(A')]$. But $det(V') = 1$, $det(U') > 0$, and $det[U(A')] = |det(A')| > 0$. Therefore, $det[V(A')] > 0$, and so we must have $det[V(A')] = 1$ and $V(A')$ is a rotation. Since both V' and $V(A')$ are rotations, the product $(V')^t V(A')$ is also a rotation. Hence U' is a rotation matrix times $U(A')$. The fact that both of the latter are upper triangular with positive diagonal entries forces $(V')^t V(A') = I$. That being the case, one must have $V(A') = V'$ and $U(A') = U'$. §

Proposition 2 shows that the conditions $V' \in \mathcal{V}$ and $U' \in \mathcal{U}$ are sufficient to guarantee the stated result. It is easy to see that they are necessary as well. Proposition 2 says, in effect, that the factorization into matrices belonging to the defined sets is unique.

Proposition 3.
Suppose $\Lambda' > 0$, $V' \in \mathcal{V}$, and $S' \in \mathcal{S}$. If $A' = \Lambda'\, V'\, S'$, then $\Lambda(A') = \Lambda'$, $V(A') = V'$ and $S(A') = S'$.
Proof.
The proof for $V' \in \mathcal{R}$ is provided, while the case $V' \in \mathcal{F}$ is similar. Note first that, $det(A') = (\Lambda')^2\, det(V')\, det(S')$. Because $V' \in \mathcal{V}$, $det(V') = 1$. In addition, $S' \in \mathcal{S}$ gives $det(S') = 1$. Therefore, $det(A') = (\Lambda')^2 > 0$. In turn, this means $A' \in M_2^*$ and so the the factorization $A' = \Lambda(A')\, V(A')\, S(A')$ exists. Thus, we have

$$\Lambda'\, V'\, S' \quad = \quad \Lambda(A')\, V(A')\, S(A')$$

By definition, $\Lambda(A')\, S(A') = U(A')$. Further, if we let $U' = \Lambda'\, S'$, we see that U' is upper triangular with positive diagonal entries, so $U' \in \mathcal{U}$. Thus, $A' = V'\, U'$ and Proposition 2 can be applied to show $V' = V(A')$ and $U' = U(A')$. The latter gives

$$\Lambda'\, S' = \Lambda(A')\, S(A')$$

Taking the determinant of both sides of the expression and equating, we find that, $[\Lambda']^2 = [\Lambda(A')]^2$. But since both of these scalars are positive, that means $\Lambda' = \Lambda(A')$. Then, $S' = S(A')$ is immediate. §

The assumptions in Proposition 3 are both sufficient and necessary to guarantee the stated results.

Proposition 4.
Suppose $\Lambda' > 0$, $V' \in \mathcal{V}$, $Q' \in \mathcal{Q}$, and $\Delta' \in \mathcal{D}$. If $A' = \Lambda'\, V'\, Q'\, \Delta'$, then $\Lambda(A') = \Lambda'$, $V(A') = V'$, $Q(A') = Q'$, and $\Delta(A') = \Delta'$.
Proof.

The proof for $V' \in \mathcal{R}$ is provided, while the case $V' \in \mathcal{F}$ is similar. We have $det(A') = (\Lambda')^2$ because $V' \in \mathcal{V}$, $Q' \in \mathcal{Q}$, and $\Delta' \in \mathcal{D}$ requires that these matrices have unit determinant. Thus A' is non-singular and the factorization exists, i.e., $A' = \Lambda(A')\, V(A')\, Q(A')\, D(A')$. Therefore

$$\Lambda'\, V'\, Q'\, \Delta' \quad = \quad \Lambda(A')\, V(A')\, Q(A')\, \Delta(A')$$

By definition, $Q(A')\, \Delta(A') = S(A')$. Further, if we let $S' = Q'\, \Delta'$, we see that $S' \in \mathcal{S}$. Thus, $A' = \Lambda'\, V'\, S'$ and Proposition 3 can be applied to show $\Lambda(A') = \Lambda'$, $V(A') = V'$, and $S(A') = S'$. The latter gives

$$Q'\, \Delta' = Q(A')\, \Delta(A')$$

Multiplying the matrices on each side of this relation and equating the elements reveals that $Q' = Q(A')\, diag(r, 1/r)$ for some $r > 0$. But since the columns of Q' have equal length, so must the columns of the product $Q(A')\, diag(r, 1/r)$. This forces $r = 1$, and thus $diag(r, 1/r)$ is the identity matrix. Therefore, $Q(A') = Q'$. It is then immediate that $\Delta(A') = \Delta'$. §

The assumptions in Proposition 4 are both sufficient and necessary to guarantee the stated results.

Corollary.
Let $G_k \in M_2^*$ with $k = 1, 2, 3, 4$, and let $G = \Lambda(G_1)\, V(G_2)\, Q(G_3)\, \Delta(G_4)$. Then

$$\begin{aligned} \Lambda(G) &= \Lambda(G_1) \\ V(G) &= V(G_2) \\ Q(G) &= Q(G_3) \\ \Delta(G) &= \Delta(G_4) \end{aligned}$$

Proof.
From prior observations, $\Lambda(G_1) > 0$, $V(G_2) \in \mathcal{V}$, $Q(G_3) \in \mathcal{Q}$, and $\Delta(G_4) \in \mathcal{D}$. Applying Proposition 4, we have the stated result. §

Similarly corollaries may be proved using Propositions 2 and 3. For example, if $G_1 \in M_2^*$, $G_2 \in M_2^*$, and $G = V(G_1)\, U(G_2)$, then $V(G) = V(G_1)$ and $U(G) = U(G_2)$.

4 Target-matrix Construction Using the 2×2 Factorization

The propositions in the previous section justify the following approach to Target-matrix construction. Given an initial mesh, a set of matrices A_{init} can be created from the underlying mappings. If $A_{init} \in M_2^*$, then from the expressions (4)-(9), initial values $V_{init} = V(A_{init})$, $U_{init} = U(A_{init})$, etc., can be found such that

$$\begin{aligned} A_{init} &= V_{init}\, U_{init} \\ &= \Lambda_{init}\, V_{init}\, S_{init} \\ &= \Lambda_{init}\, V_{init}\, Q_{init}\, \Delta_{init} \end{aligned}$$

To construct Target-matrices from these factorizations, one or more of these initial matrix factors can be replaced with a new matrix. For example, if we set $W = \Lambda_{init}\, V_{init}\, S_{new}$, then according to the propositions, $\Lambda(W) = \Lambda_{init}$, $V(W) = V_{init}$, and $S(W) = S_{new}$, provided $S_{new} \in \mathcal{S}$. If this target were used along with the local metric $|T - I|^2$, optimization should produce an optimal mesh whose shape quality is now close to S_{new}, while the Size and Orientation qualities would remain close to those of the initial mesh.

It is clear that many different combinations of *new* and *initial* matrices can be created and combined to form a target matrix. For the moment, let us leave aside the question of what combinations exist, and which are important, to focus instead on how one might create useful λ_{new}, V_{new}, Q_{new}, Δ_{new}, S_{new}, and U_{new} matrices.

The *initial* matrices V_{init}, etc., will in general vary from one sample point in the mesh to another. To realize the full potential of the Target-matrix paradigm, the *new* matrices V_{new}, etc., must also be allowed to vary over the sample points. Let $(V_{new})_k$ denote the new Orientation matrix at sample point k, and use similar notation for the other matrices.

The following subsections provide high-level methods for creating *new* matrices that belong to the proper sets $\mathcal{V}$, etc., so that the Propositions of the previous section hold. Notably, *the matrix factors are created from data available to the application.* This data, of course, varies from one application to the next. Most prior methods of mesh optimization do not use the data to be described below, at least not in such a systematic fashion. Thus, target construction based on this seldom-used data potentially permits the creation of better optimal meshes. Of course, Target-matrix creation is an art because the ingenuity of the creator plays an important role in the quality of the results.

4.1 The Size Factor $(\Lambda_{new})_k$

Recall that the Size factor $\Lambda(A) = \sqrt{|\alpha|}$ is related to the local area of the map. Clearly, if the application desires a relatively large element in some location, then Λ should be relatively large. The area, in turn, depends on the lengths and included angle between the local tangent vectors. If the angle ϕ is controlled via the Q-matrix, so that $\sin\phi \approx 1$, then Λ roughly controls the geometric average of the lengths of the tangents. These, in turn, are related to the lengths of the edges in an element.

A generic model for calculating the Size factor over the set of sample points in the mesh assumes that we are given application data such that a set of *positive* scalars $\{f_k\}$ defined at the sample points can be computed. If there are K sample points in the mesh, let $\bar{f} = \frac{1}{K}\sum_k f_k$ be the average value over the set of values. Further, given $(\Lambda_{init})_k$ as the value of Λ at each sample point of the initial mesh, let $\bar{\Lambda} = \frac{1}{K}(\Lambda_{init})_k$ be the average Size in the initial mesh. A properly-scaled model for a new Size factor is

$$(\Lambda_{new})_k = \left(\frac{f_k}{\bar{f}}\right)\bar{\Lambda} \tag{10}$$

This model will tend to create Sizes in the optimal mesh that are proportional to f_k. Note that this scaling is such that $\frac{1}{K}\sum_k(\Lambda_{new})_k = \bar{\Lambda}$, so that the average Λ in the optimal mesh is the same as in the initial mesh. This property helps ensure the mesh remains of good quality. The requirement that f be a positive function comes from the fact that we need Λ positive in order for the factorization Propositions to hold. When $f_k \equiv 1$ over all the sample points, we have the special case $(\Lambda_{new})_k = \bar{\Lambda}$, which corresponds to the goal of creating a mesh with equal Size everywhere.[7]

If Size is important to control while optimizing the mesh, then of course, one must not use a size-invariant local metric such as the Shape metric. The Size+Shape metric or the Size+Shape+Orientation metrics are logical candidates.

Applications may have available the following data which can be used to create useful $\{f_k\}$:

- a posteriori discretization error estimates or error indicator data,
- scalars derived from physically-meaningful scalar fields such as conductivity, permeability, depth contours, etc.,

[7]Another special case is obtained by letting $f_k = (\Lambda_{init})_k$, which gives $(\Lambda_{new})_k = (\Lambda_{init})_k$.

- determinants of symmetric positive definite matrices such as a the solution Hessian, the thermal conductivity, the Permeability tensor, etc.,
- the norm of the solution gradient, or
- surface curvature.

It is evident from these examples that a wide variety of application-specific Size-adapted optimal meshes can potentially be created.

4.2 The Orientation Factor $(V_{new})_k$

Recall that the Orientation matrix V is related to the angle θ that the first column of A makes with the x-axis. If the angle ϕ is controlled by the Q-matrix, then V additionally controls the local orientation of the tangent vectors to the map with respect to the global coordinate system.

A generic model for calculating a new Orientation matrix to be used in the Target assumes that we are given application data from which a set of *non-zero* vectors $\{(u_k, v_k)\}$ defined at the sample points can be computed. Let $\cos\theta_k = u_k/\sqrt{u_k^2+v_k^2}$ and $\sin\theta_k = v_k/\sqrt{u_k^2+v_k^2}$, so that

$$(V_{new})_k = \frac{1}{\sqrt{u_k^2+v_k^2}} \begin{pmatrix} u_k & -v_k \\ v_k & u_k \end{pmatrix} \tag{11}$$

With this construction, V_{new} is a rotation, and the optimal mesh should tend to align the first column of the Jacobian matrix of the optimized mesh with the vector field at each sample point.

In some cases, one may wish to align the *second* column of the Jacobian matrix with the vector field. To do so, we propose

$$(V_{new})_k = \frac{1}{\sqrt{u_k^2+v_k^2}} \begin{pmatrix} v_k & u_k \\ -u_k & v_k \end{pmatrix} \tag{12}$$

If Orientation is important to control while optimizing the mesh, then of course, one must not use an orientation-invariant local metric such as the Shape or Size+Shape metrics. The Size+Shape+Orientation metric is the proper choice.

Applications may have available, among other possibilities, the following data which can be used to create useful $\{(u_k, v_k)\}$: magnetic fields, fluid velocity vectors, solution gradients, fluxes, or eigenvectors of symmetric tensor fields. It is evident from these examples that a wide variety of application-specific Orientation-adapted optimal meshes can potentially be created.

This approach was originally suggested in [11], prior to the TMP.

It is important to note that, to use this approach effectively requires knowledge of the way in which the vertices of each mesh element are labeled. An example is given in Figure 3. In the terminology of [5], V is not a label-invariant quantity, whereas Λ is. The issue is not a major difficulty for structured meshes since the labeling is often consistent from one element to the next, but can be troublesome on unstructured meshes because an incorrect assumption on the labeling could result in a tangled mesh. Moreover, in practice, orientation-control seems to be less commonly wanted with unstructured meshes because such meshes do not have global tangent lines.

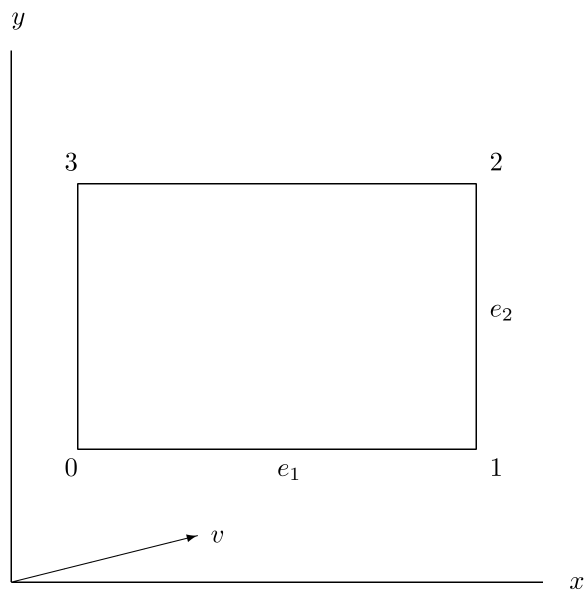

Figure 3: A sample point located at vertex 0 of the element shown has the first column of A equal to e_1 assuming the labeling shown. However, if the labeling is cyclically permuted, so that 0123 becomes 3012, then the sample point at vertex 0 has the first column of A equal to e_2. Thus, depending on the labeling, either e_1 or e_2 would align with v.

4.3 The Skew Factor $(Q_{new})_k$

Recall that the Skew-matrix is related to the angle ϕ between the two tangent vectors of the map. Assume that we are given application data such that the set of angles $\{\phi_k\}$ (with $\sin\phi_k > 0$) over all sample points can be computed. The generic model for calculating a new Skew-matrix is based directly on the geometric definition of Q given in the previous section:

$$(Q_{new})_k \quad = \quad \frac{1}{\sqrt{\sin\phi_k}} \begin{pmatrix} 1 & \cos\phi_k \\ 0 & \sin\phi_k \end{pmatrix} \tag{13}$$

The condition on $\sin\phi_k$ helps meet the conditions for the factorization Propositions to hold.

In practice, many applications may not be able to supply meaningful data from which we can calculate the set $\{\phi_k\}$, varying over the sample points. One remedy is simply to set ϕ_k to a constant angle which is related to the angle found in the ideal element type. For example, the ideal quadrilateral element has $\phi = \pi/2$, which results in $(Q_{new})_k = I$. The ideal triangle element has $\phi = \pi/3$, giving

$$(Q_{new})_k = \frac{1}{\sqrt{2}\,\sqrt[4]{3}} \begin{pmatrix} 2 & 1 \\ 0 & \sqrt{3} \end{pmatrix} \tag{14}$$

The use of constant Q matrices in the Target is appropriate if a goal of the optimization is to create a mesh whose local tangents all include the ideal angle.

None of the local metrics in the Target-matrix paradigm are invariant to the skew-angle, so Q_{new} can be used with any metric.

4.4 The Aspect Ratio Factor $(\Delta_{new})_k$

Recall that the Δ-matrix is related to the local tangent aspect ratio $\sqrt{\frac{r_1}{r_2}}$. A generic model for calculating the Aspect Ratio matrix over the set of sample points in the mesh assumes that we are given application data such that a set of *positive* scalars $\{\rho_k\}$ defined at the sample points can be computed. Then let

$$(\Delta_{new})_k = \begin{pmatrix} \sqrt{\rho_k} & 0 \\ 0 & 1/\sqrt{\rho_k} \end{pmatrix} \tag{15}$$

An important special case occurs with $\rho_k \equiv 1$ for all k. Then $(\Delta_{new})_k = I$, which is appropriate if one desires that the optimal mesh be isotropic.

None of the local metrics in the Target-matrix paradigm are invariant to aspect ratio, so Δ_{new} can be used with any metric.

Applications may have available, among other possibilities, the following data which can be used to create useful values of ρ_k: the eigenvalues of a symmetric positive definite matrix such as the Hessian, the Curvature, or the Diffusivity, a priori knowledge such as the aspect ratio of the geometric domain or the aspect ratio of a boundary layer, or pairs of vectors such as the electric and magnetic field vectors.

As with Orientation, Aspect-Ratio is not label-invariant. Thus, in Figure 3, the aspect-ratio at vertex 0 is r_1/r_2 is e_1/e_2 when the labeling of the element vertices is as shown, whereas, the aspect-ratio at vertex 0 becomes e_2/e_1 under the label permutation $0123 \to 3012$. Therefore, to use this approach to aspect-ratio effectively requires specific knowledge of the way in which the vertices of each mesh element is labeled.

4.5 The Shape and Shape+Size Factors $(S_{new})_k$, $(U_{new})_k$

Recall that the Shape matrix is related to both angle and aspect ratio. A generic model for calculating a new Shape matrix is to take advantage of the relation $S_{new} = Q_{new}\Delta_{new}$. Therefore, a new Shape matrix is found by first calculating new Skew and Aspect Ratio matrices according to the previous two subsections, and then multiplying them together.

The same approach can be used to calculate a new Size+Shape matrix $U_{new} = \Lambda_{new} S_{new}$.

4.6 Creating *New* Factors by Multiplying *Initial* Factors

In the preceding subsections, we presented an approach to Target construction in which *new* factors, created from application data, are substituted for *initial* factors. The *initial* factors are thus discarded. *New* factors can also be created by modification of the *initial* factors. This may be a more convenient way to construct Targets in certain situations.

The basic idea is the following: let $\lambda_k > 0$, $V_k \in \mathcal{V}$, $D_k \in \mathcal{D}$ be calculated from data supplied by the application, for all k. Then define *new* factors in terms of the *initial* factors as follows:

$$\begin{aligned} (\Lambda_{new})_k &= \lambda_k\, (\Lambda_{init})_k \\ (V_{new})_k &= V_k\, (V_{init})_k \\ (\Delta_{new})_k &= (\Delta_{init})_k\, D_k \end{aligned}$$

where $\Lambda_{init} = \Lambda(A_{init})$, $V_{init} = V(A_{init})$, and $\Delta_{init} = \Delta(A_{init})$. These operations are possible because the sets $\mathcal{V}$ and $\mathcal{D}$ are closed under multiplication.[8] The new factors then replace the initial factors, as before. For example, if one desired $W = \Lambda_{init}\, V_{new}\, Q_{new}\, \Delta_{init}$, then this method of creating *new* factors gives

$$\begin{aligned} W_k &= (\Lambda_{init})_k\, (V_{new})_k\, (Q_{new})_k\, (\Delta_{init})_k \\ &= (\Lambda_{init})_k\, V_k\, (V_{init})_k\, (Q_{new})_k\, (\Delta_{init})_k \end{aligned}$$

[8] The sets $\mathcal{S}$ and $\mathcal{U}$ are also closed, so a similar approach can be taken to create S_{new} or U_{new} from S_{init} or U_{init}. However, the set $\mathcal{Q}$ is not closed under multiplication, so this approach cannot be taken with the Skew factor.

with $V(W_k) = V_k\,(V_{init})_k$. One can also create a *new* Jacobian matrix from the *initial* Jacobian matrix by letting

$$(A_{new})_k \quad = \quad \lambda_k\, V_k\, (A_{init})_k\, D_k$$

This would be used in the Target $W_k = (A_{new})_k$. The construction can be written

$$(A_{new})_k \quad = \quad [\lambda_k\, (\Lambda_{init})_k]\; [V_k\, (V_{init})_k]\; (Q_{init})_k\; [(\Delta_{init})_k\, D_k]$$

Because the sets are closed, the Propositions allow us to conclude that

$$\begin{aligned} \Lambda[(A_{new})_k] &= \lambda_k\, (\Lambda_{init})_k \\ V[(A_{new})_k] &= V_k\, (V_{init})_k \\ \Delta[(A_{new})_k] &= (\Delta_{init})_k\, D_k \end{aligned}$$

Usually, V_k will be a rotation, but if, at a particular sample point, $(V_{init})_k$ is a flip, then perhaps choosing V_k to be a flip might be effective in eliminating an inverted element. In fact, this choice is necessary since we require that all Target-matrices have positive determinant.

If a reference mesh is available, then one could use $\lambda_k = \Lambda(A_{ref})$, $V_k = V(A_{ref})$, and/or $D_k = \Delta(A_{ref})$.

It is unclear if this approach to creating new factors by modifying the initial factors is useful in practice. In a later section, we see that cases involving one or more of λ_k, V_k, or D_k being constant could be of use in mesh Morphing or Copying.

4.7 Closing Remarks on the Factorization Approach to Target Construction

In summary, given appropriate application data, the generic models of the previous subsections can be used to compute *new* matrix factors to be used in combination with the *initial* matrix factors to construct Target-matrices over the set of sample points such that a particular mesh optimization goal is met. A Target combination such as $W = \Lambda_{init}\, V_{new}\, Q_{new}\, \Delta_{init}$, for example, when used with the Size+Shape+Orientation local metric, essentially says that the goal is to *preserve* the initial mesh qualities related to Size and Aspect Ratio, while *improving* the qualities related to Orientation and Skew. Thus, in TMP, one can, as a goal, both preserve and improve various mesh qualities via optimization.

As noted Section Two, it is not necessary that the set of Target-matrices constructed from the factorization method be consistent from one sample point to another because the least-squares objective function will reconcile the inconsistencies. However, the more consistent the set of Targets is, the more likely the optimal mesh will satisfy the optimization goal implied by the targets.

Two additional methods for creating *new* matrix factors should be mentioned.

First, if a reference mesh is available, one can compute the factorization $A_{ref} = \Lambda_{ref}\, V_{ref}\, Q_{ref}\, \Delta_{ref}$. Then one can let, for example, $\Lambda_{new} = \Lambda_{ref}$, $V_{new} = V_{ref}$, etc. Hence, one can have Target-matrices with combinations such as $W = \Lambda_{ref}\, V_{init}\, S_{new}$ in which the target is composed of information from the reference mesh, the initial mesh, and from application-specific data. This example would make sense if the reference mesh has good Size quality, but poor Shape quality.

Second, a similar approach can be taken that involves a two-stage optimization procedure in which the optimal mesh from the first stage is used to create targets for the second stage. For example, an optimal mesh could first be created by successively optimizing local patches from the initial mesh using a Shape metric with target based on the ideal element shape. After each local patch has been optimized one calculates $(A_{opt})_k$. The optimization continues to the next local patch, but not before restoring the coordinates of the center vertex from the optimized local patch to its initial position. This is followed by a subsequent global optimization, again beginning with the initial mesh, but using the Size+Shape+Orientation metric, and the target $W = A_{opt}$ to create a mesh that is similar to the initial mesh, but with better shape quality. This method was proposed in the ALE rezone paper [12]. Other possibilities involving factorization of the Jacobian matrix of optimal mesh resulting from the first optimization stage to obtain *new* matrix factors that can be inserted into the second-stage target-matrices may be useful.

In principle, there is nothing to prevent one from varying the combination of *initial* and *new* matrix factors from one sample point to the next. For example, the sample points in one subset of the mesh elements might use $W = A_{init}$, while another subset might use $W = A_{new}$. Such a possibility raises the issue of Target-matrix smoothness. Numerical experiments in the past suggest that if the Targets do not vary smoothly over the set of sample points, the optimal mesh will not be smooth.[9] A combination of *initial* and *new* targets is likely to be non-smooth unless carefully done, so this possibility may be unattractive for that reason. The same can be said concerning the matrix factors, i.e., they should vary smoothly over the sample points if a smooth optimal mesh is wanted.

5 Extension of the Factorization to Three Dimensions

Sections 3 and 4 focused on 2×2 matrices. In this section the results are extended to 3×3 matrices, which are associated with volume mesh elements.

Let M_3 be the set of all 3×3 matrices with real elements, and let M_3^* be the set of non-singular matrices in M_3. Consider $A_{3\times 3}$, with the columns of A given by the vectors a_1, a_2, and a_3, and write $A = [a_1, a_2, a_3]$. If $A \in M_3^*$, then $\alpha \equiv det(A) \neq 0$ and thus the quantities $|a_1|$, $|a_2|$, $|a_3|$, and $|a_1 \times a_2|$ are non-zero. When A is the Jacobian matrix, the three column vectors correspond to the three tangent vectors of the mapping (at the given sample point).

For $A \in M_3^*$, the factorizations (1)-(3) still hold, provided

$$\Lambda(A) = |\alpha|^{1/3} \tag{16}$$

$$V(A) = \left(\frac{a_1}{|a_1|}, \frac{-(a_1 \cdot a_2)a_1 + |a_1|^2 a_2}{|a_1||a_1 \times a_2|}, \frac{\alpha(a_1 \times a_2)}{|\alpha||a_1 \times a_2|} \right) \tag{17}$$

$$Q(A) = \sqrt[3]{\frac{|a_1||a_2||a_3|}{|\alpha|}} \begin{pmatrix} 1 & \frac{a_1 \cdot a_2}{|a_1||a_2|} & \frac{a_1 \cdot a_3}{|a_1||a_3|} \\ 0 & \frac{|a_1 \times a_2|}{|a_1||a_2|} & \frac{(a_1 \times a_2)\cdot(a_1 \times a_3)}{|a_1 \times a_2||a_1||a_3|} \\ 0 & 0 & \frac{|\alpha|}{|a_1 \times a_2||a_3|} \end{pmatrix} \tag{18}$$

$$\Delta(A) = \frac{1}{\sqrt[3]{|a_1||a_2||a_3|}} \begin{pmatrix} |a_1| & 0 & 0 \\ 0 & |a_2| & 0 \\ 0 & 0 & |a_3| \end{pmatrix} \tag{19}$$

$$S(A) = \frac{1}{\sqrt[3]{|\alpha|}} \begin{pmatrix} |a_1| & \frac{a_1 \cdot a_2}{|a_1|} & \frac{a_1 \cdot a_3}{|a_1|} \\ 0 & \frac{|a_1 \times a_2|}{|a_1|} & \frac{(a_1 \times a_2)\cdot(a_1 \times a_3)}{|a_1 \times a_2||a_1|} \\ 0 & 0 & \frac{|\alpha|}{|a_1 \times a_2|} \end{pmatrix} \tag{20}$$

$$U(A) = \begin{pmatrix} |a_1| & \frac{a_1 \cdot a_2}{|a_1|} & \frac{a_1 \cdot a_3}{|a_1|} \\ 0 & \frac{|a_1 \times a_2|}{|a_1|} & \frac{(a_1 \times a_2)\cdot(a_1 \times a_3)}{|a_1 \times a_2||a_1|} \\ 0 & 0 & \frac{|\alpha|}{|a_1 \times a_2|} \end{pmatrix} \tag{21}$$

[9] The experiments also suggest that, while target-smoothness is necessary for a smooth optimal mesh, it may not be sufficient.

The matrices above belong to the sets $\mathcal{V}$, $\mathcal{U}$, $\mathcal{S}$, $\mathcal{Q}$, and $\mathcal{D}$ defined in Section 3, but now on M_3^*.[10]

The factors of $A = [a_1, a_2, a_3]$ again have recognizable geometric meanings. Let $r_i = |a_i|$ and ψ_{ij}, $(i,j) \in \{1,2,3\}$, be the included angle between a_i and a_j, and further assume that for $j \neq i$, $0 < \psi_{ij} < \pi$ so that the determinant of A will be positive. Let ν be the angle between the vectors $a_1 \times a_2$ and $a_1 \times a_3$. Then one can show

$$\begin{aligned}
\cos\nu &= \frac{\cos\psi_{23} - \cos\psi_{12}\cos\psi_{13}}{\sin\psi_{12}\,\sin\psi_{13}} \\
\sin\nu &= \frac{\alpha}{r_1\, r_2\, r_3\, \sin\psi_{12}\, \sin\psi_{13}}
\end{aligned}$$

Then the factorization above becomes

$$\begin{aligned}
\Lambda(A) &= |\alpha|^{1/3} \\
V(A) &= \frac{1}{\sin\psi_{12}}\left(\frac{a_1}{r_1}\sin\psi_{12}, \frac{a_2}{r_2} - \frac{a_1}{r_1}\cos\psi_{12}, \frac{a_1}{r_1} \times \frac{a_2}{r_2}\right) \\
Q(A) &= \sqrt[3]{\frac{r_1\, r_2\, r_3}{|\alpha|}} \begin{pmatrix} 1 & \cos\psi_{12} & \cos\psi_{13} \\ 0 & \sin\psi_{12} & \sin\psi_{13}\,\cos\nu \\ 0 & 0 & \sin\psi_{13}\,\sin\nu \end{pmatrix} \\
\Delta(A) &= \frac{1}{\sqrt[3]{r_1\, r_2\, r_3}} \begin{pmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{pmatrix} \\
S(A) &= \frac{1}{\sqrt[3]{|\alpha|}} \begin{pmatrix} r_1 & r_2\,\cos\psi_{12} & r_3\,\cos\psi_{13} \\ 0 & r_2\,\sin\psi_{12} & r_3\,\sin\psi_{13}\,\cos\nu \\ 0 & 0 & r_3\,\sin\psi_{13}\,\sin\nu \end{pmatrix} \\
U(A) &= \begin{pmatrix} r_1 & r_2\,\cos\psi_{12} & r_3\,\cos\psi_{13} \\ 0 & r_2\,\sin\psi_{12} & r_3\,\sin\psi_{13}\,\cos\nu \\ 0 & 0 & r_3\,\sin\psi_{13}\,\sin\nu \end{pmatrix}
\end{aligned}$$

As in the 2D case, the quantities Λ, V, Q, and Δ can be seen to be related to size, orientation, skew, and aspect ratio, respectively.

Proposition 1 from Section 3 holds for $d = 3$ as well as $d = 2$, as can be seen from a re-reading of the proof. Changing M_2^* to M_3^* in the proof of Proposition 2 gives a valid proof that the proposition also holds for $d = 3$. With minor modifications to the proofs of Propositions 3, 4, and 5, one can show that these also hold for $d = 3$.

6 Target-matrix Construction using the 3×3 Factorization

The main difference between this section and Section 4 is due to the different explicit $\Lambda V Q \Delta$ factorizations for 2×2 and 3×3 matrices. Thus, the emphasis in this section is on a description of the generic models for the *new* matrices in the factorization of 3×3 Jacobian matrices. The models rely heavily on the expressions given at the end of the previous section. The approach described in subsection 4.6 carries over immediately to the three-dimensional case.

6.1 The Size Factor $(\Lambda_{new})_k$

The Size factor when $d = 3$ is related to the local volume of the map at a sample point. The generic model is identical to that given in Section 4, as are the sources for the application

[10] The identity $|\alpha||a_1| \equiv |(a_1 \times a_2) \times (a_1 \times a_3)|$ is useful in showing that the lengths of the columns of Q are equal.

data used to determine $\{f_k\}$.

6.2 The 3×3 Orientation Factor $(V_{new})_k$

To determine a new 3×3 orientation matrix requires *two* non-zero vectors $\{\mathbf{u}_k\}$ $\{\mathbf{v}_k\}$ at each sample point, with $|\mathbf{u}_k \times \mathbf{v}_k| \neq 0$. Then the generic model is the rotation

$$(V_{new})_k = \left[\frac{\mathbf{u}_k}{|\mathbf{u}_k|}, \frac{|\mathbf{u}_k|^2 \, \mathbf{v}_k - (\mathbf{u}_k \cdot \mathbf{v}_k)\, \mathbf{u}_k}{|\mathbf{u}_k| \, |\mathbf{u}_k \times \mathbf{v}_k|}, \frac{\mathbf{u}_k \times \mathbf{v}_k}{|\mathbf{u}_k \times \mathbf{v}_k|} \right]$$

This will tend to make the optimal mesh align with the two vector fields such that the first column of the Jacobian will align with $\mathbf{u}$ and the second column with $\mathbf{v}$.

To align the *second* and *third* columns of the active Jacobian matrix with the pair of vectors $\mathbf{u}$ and $\mathbf{v}$, respectively, let

$$(V_{new})_k = \left[\frac{\mathbf{u}_k \times \mathbf{v}_k}{|\mathbf{u}_k \times \mathbf{v}_k|}, \frac{\mathbf{u}_k}{|\mathbf{u}_k|}, \frac{|\mathbf{u}_k|^2 \, \mathbf{v}_k - (\mathbf{u}_k \cdot \mathbf{v}_k)\, \mathbf{u}_k}{|\mathbf{u}_k| \, |\mathbf{u}_k \times \mathbf{v}_k|} \right]$$

and to align the *third* and *first* columns of the active Jacobian matrix with the pair of vectors $\mathbf{u}$ and $\mathbf{v}$, respectively, let

$$(V_{new})_k = \left[\frac{|\mathbf{u}_k|^2 \, \mathbf{v}_k - (\mathbf{u}_k \cdot \mathbf{v}_k)\, \mathbf{u}_k}{|\mathbf{u}_k| \, |\mathbf{u}_k \times \mathbf{v}_k|}, \frac{\mathbf{u}_k \times \mathbf{v}_k}{|\mathbf{u}_k \times \mathbf{v}_k|}, \frac{\mathbf{u}_k}{|\mathbf{u}_k|} \right]$$

Applications may have available, among other possibilities, the following data which can be used to create useful pairs $\mathbf{u}_k$, $\mathbf{v}_k$: electric and magnetic fields, or eigenvectors of matrices. In practice, the application may have access to only one vector field, so some ingenuity will be required to find a suitable second field.

$V_{3\times3}$ is not a label-invariant quantity, so some effort is needed to properly correlate element vertex numberings with the desired alignment.

6.3 The 3×3 Skew Factor $(Q_{new})_k$

The angles ψ_{ij} refer to the angles between the vector triple consisting of the columns of the Jacobian matrix at a sample point. A generic model for $(Q_{new})_k$ could be based on these angles, according to the Skew matrix factor of the previous section. In practice, few applications will be able to supply these angles at all the sample points. More useful perhaps is the case where Q_{new} is a constant which is based on the angles in the ideal element. For example, in a hexahedral element, the face angles are all $\pi/2$ so that $(Q_{new})_k = I$ at all the sample points. The use of constant Skew matrices in the Target is appropriate if a goal of the optimization is to create a mesh whose local tangents all include the ideal angle.

For a tetrahedral element, the face angles are all $\pi/3$, giving

$$(Q_{new})_k = \sqrt[6]{2} \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{6} \\ 0 & 0 & \sqrt{\frac{2}{3}} \end{pmatrix}$$

For a triangular wedge element, with face angles $\psi_{12} = \frac{\pi}{3}$ and $\psi_{23} = \psi_{13} = \frac{\pi}{2}$,

$$(Q_{new})_k = \begin{pmatrix} 1 & \frac{1}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

For a square pyramid element of height h and base edge-length ℓ, let

$$r = \sqrt{1 + \frac{1}{2}\left(\frac{\ell}{h}\right)^2}$$

and

$$(Q_{new})_k = \sqrt[3]{r}\begin{pmatrix} 1 & 0 & \frac{\ell}{2rh} \\ 0 & 1 & \frac{\ell}{2rh} \\ 0 & 0 & \frac{1}{r} \end{pmatrix}$$

6.4 Aspect Ratio Schemes $(\Delta_{new})_k$

A generic model for Δ requires that we be given three sets of positive numbers $\{(r_1)_k\}$, $\{(r_2)_k\}$, $\{(r_3)_k\}$ at each sample point, representing relative lengths of the three tangent vectors. The model is

$$\Delta_{new} = \begin{pmatrix} \sqrt[3]{\frac{r_1^2}{r_2\,r_3}} & 0 & 0 \\ 0 & \sqrt[3]{\frac{r_2^2}{r_3\,r_1}} & 0 \\ 0 & 0 & \sqrt[3]{\frac{r_3^2}{r_1\,r_2}} \end{pmatrix}$$

For an isotropic mesh, we can use $\Delta_{new} = I$, otherwise, element labeling must be taken into account in order for the results to be close to what is desired.

7 Example of Target Construction using $\Lambda V Q \Delta$

A fore-runner of the Target-matrix paradigm can be found in [10], in which a two-dimensional structured mesh based on a global map is aligned with a given vector field. The optimization is cast as a problem in the calculus of variations and the metric $|A^{-1} - W^{-1}|^2$ is used. The target was calculated directly and did not use the factorization method given in the present paper. However, the approach was similar to the construction of V_{new} given here. Many of the concepts in the present TMP were lacking in the original alignment paper, such as the factorization, the extension to finite element meshes, the weighted Jacobian matrix T, the concept of barrier metrics, and the use of Shape or Shape+Size metrics.

An example of the factorization method described in Sections 3 and 4 is given in this section. The initial mesh consists of a quadrilateral 'paved' planar region. The mesh is isotropic, with good shape quality (see Figure 4). The difficulty with this mesh is that there are a number of edges located near the domain corners which have relatively small edge-lengths. The simulation code that used the mesh is explicit in time, so these small edge-lengths control the size of the time-step via the Courant condition. The goal, then, of the optimization is to make the edge-lengths in the mesh more uniform so that a larger time-step can be taken.

The Target-matrix paradigm lends itself readily to this problem. To construct the set of Target-matrices we begin with the initial mesh, calculate the Jacobian matrix A_{init} at each sample point. The sample points consist of the four corners of each quadrilateral in the mesh. Apply the $\Lambda V Q \Delta$ factorization to obtain the factors Λ_{init}, V_{init}, Q_{init}, and Δ_{init}. The factor which most needs to be replaced with a 'new' factor is Λ_{init} because this is the primary source of the non-uniformity in edge-length. To elaborate: V_{init} controls only orientation, while Q_{init} and Δ_{init} are close to uniform since all the elements' shapes are close to being squares. Therefore, we replace Λ_{init} with Λ_{new} as given in Section 4.1. Since uniformity in edge-length is desired, we choose $f_k \equiv 1$ for all k, giving $(\Lambda_{new})_k = \bar{\Lambda}$. Although perhaps not strictly necessary, we also replace Q_{init} and D_{init} with $Q_{new} = I$ and $D_{new} = I$ in order to further improve Shape. The constructed Target-matrices are therefore

$$W_k = \bar{\Lambda}\,(V_{init})_k$$

Because element orientation is not important in this problem, V_{init} need not necessarily be preserved; therefore, we can use an orientation-invariant local metric, which allows the optimization to adjust orientation if necessary to achieve greater uniformity of edge-lengths. We cannot use a Shape metric, however, because these are insensitive to Λ. Thus, the best choice of local metric is to use a Size+Shape metric. This choice should preserve (or slightly improve) the square shape of the quadrilaterals while equidistributing the Size (and therefore, the edge-lengths).[11] Because the initial mesh is non-inverted, we can use the barrier form of the Size+Shape metric to guarantee the optimal mesh is also non-inverted. The local quality metric used to produce the optimized mesh in Figure 4 was the following barrier-based 2D Size+Shape metric:

$$\mu(T) = \frac{|T|^2 - 2\sqrt{|T|^2 + 2\tau} + 2}{2\tau}$$

From the figure one sees that indeed the optimal mesh has more uniform edge-lengths. In fact, the minimum edge-length in the optimized mesh is 1.7 times longer that the minimum edge-length in the initial mesh, thus permitting the simulation code to increase the time-step by the same factor.

The assumptions that went into the design of this optimization and Target construction method are that (1) the initial paved mesh is non-inverted (and thus a barrier-metric can be used), (2) the elements are quadrilaterals (so that $Q_{new} = I$ is appropriate), and (3) the elements are isotropic so that V_{init} need not be replaced and so that $\Delta_{new} = I$ is appropriate. These assumptions are usually reasonable when the initial mesh is obtained via a paving algorithm. Obviously, if these assumptions are violated, we would need to re-think the Target construction. For example, if the paved mesh consisted of triangular elements, a different Q_{new} is needed. Or, if the mesh is not isotropic, then perhaps a different Δ_{new} and V_{new} would be needed.

It is noted that there is some flexibility in the Target-construction algorithm for this particular problem. For example, replacing V_{init} with $V_{new} = I$ might have worked just as well since an orientation-invariant metric was used.

The most important point we wish to make is that in the Target-matrix approach, one can generally use a very small set of local metrics (Shape, Shape+Size, Shape+Size+Orientation) to address a wide variety of optimization goals. This is accomplished, not by creating new metrics, but by creating new Target-matrices.

8 Target Construction for Particular Applications

As mentioned earlier, the factorization method of Target construction allows many combinations of *initial* and *new* matrices. For example, the factorization $W = V\,U$ gives rise to four combinations: $V_{init}\,U_{init}$, $V_{init}\,U_{new}$, $V_{new}\,U_{init}$, and $V_{new}\,U_{new}$. The first is equivalent to $W = A_{init}$ and the last is equivalent to $W = A_{new}$. There are actually even more than four combinations here because either a *new* matrix can be created from a generic model (like those described in Sections 4 and 6) or it can be created from a *reference* or *optimized* mesh. Additionally, a generic model for a *new* matrix can often be created using difference types of application data such as scalars, vectors, matrices, or tensors. It is thus difficult to fully enumerate all the possible combinations resulting in Targets within

[11] A pure Size metric would also equidistribute the edge-lengths, but would not necessarily preserve the square shape of the elements.

TMP. Beyond that, there are three local quality metrics, Shape (Sh), Size+Shape (SS), Size+Shape+Orientation (SSO), that can be associated with the different target combinations, leading to a large number of possible optimization problems. It turns out that many of the possible combinations are not particularly useful, so a better approach than direct enumeration is needed to find useful combinations.

Instead of trying to enumerate all possible combinations in Target construction, we first list (at an abstract level) the possible uses of each of the four factors in the $\Lambda V Q \Delta$ factorization, beginning with the Size factor, Λ.

The distribution of Size values over the initial and the optimal mesh may be either homogeneous (constant) or heterogeneous (variable). Except for some special cases, most meshes do not have a Size distribution which is exactly constant. However, if the Size is nearly constant (i.e., the Size distribution has small standard deviation), we shall call the Size homogeneous. Four uses of the size factor are identified below:

Sz_1. Create a homogeneous Size-distribution (Λ_{new} = constant),

Sz_2. Preserve the Size-distribution of the initial mesh (Λ_{init}),

Sz_3. Create a particular heterogeneous Size-distribution based on application data (Λ_{new}), or

Sz_4. Indifference to size distribution (Λ_{init} or $\Lambda = 1$).

One is rarely indifferent to the Size distribution in a mesh. However, we list Sz4 anyway because the TMP Shape metrics are invariant to Size, and experience has shown that use of the Shape metric generally does not dramatically alter whatever Size distribution is found in the initial mesh. There are certain applications in which we can take advantage of this situation which are described later in this section. Thus, Sz_4 requires the Shape metric, while Sz_1-Sz_3 can be used with either the Size-Shape metric or the Size-Shape-Orientation metric.

We can similarly list three uses of the Orientation factor:

Or_1. Create a particular mesh Orientation (V_{new}),

Or_2. Preserve the Orientation of the initial mesh (V_{init}), or

Or_3. Indifference to Orientation (V_{init} or $V = I$).

Indifference to orientation is quite common in applications of mesh optimization, particularly if the mesh is isotropic. When one is not indifferent to orientation, one is generally trying to align the mesh elements with some feature of the solution to a simulation. Many times alignment can be achieved simply by conforming the mesh to the boundary of a geometric domain. However, if an Orientation different than that implied by the geometry is wanted, mesh optimization using V_{new} can play a useful role. Or_1-Or_2 require the Size+Shape+Orientation metric, while Or_3 can be used with either Shape or Size+Shape metrics.

We list three uses of both the Skew and the Aspect Ratio factors.

Sk_1. Create a constant Skew based on element type (Q_{new} = ideal),

Sk_2. Preserve the Skew distribution of the initial mesh (Q_{init}), or

Sk_3. Create a particular non-constant Skew distribution (Q_{new}).

Constant skew means that the Skew-matrix is constant over the set of sample points within mesh elements of the same type. A hybrid mesh can have constant Skew, meaning that, for example, all the sample points within triangular elements in the mesh use the Skew-matrix in equation (21), while all the sample points within quadrilateral elements use $Q = I$.

Table 1: Potential Applications of the Target-matrix Paradigm

App	Sz-Goal	Data	Or-Goal	Data	AR-Goal	Data	Metric
1	Sz_4	initial	Or_3	initial	AR_1	ideal	Sh
2	Sz_4	initial	Or_3	initial	AR_2	initial	Sh
3	Sz_4	reference	Or_3	reference	AR_3	$\Delta(A_{ref})\,D$	Sh
4	Sz_1	constant	Or_3	initial	AR_1	ideal	SS
5	Sz_1	constant	Or_3	initial	AR_2	initial	SS
6	Sz_2	initial	Or_3	initial	AR_1	ideal	SS
7	Sz_2	initial	Or_3	initial	AR_2	initial	SS
8	Sz_3	curvature	Or_3	initial	AR_1	ideal	SS
9	Sz_3	error	Or_3	initial	AR_1	ideal	SS
10	Sz_3	$\lambda\,\Lambda(A_{ref})$	Or_3	reference	AR_3	$\Delta(A_{ref})\,D$	SS
11	Sz_3	tensor	Or_1	tensor	AR_3	tensor	SSO
12	Sz_1	constant	Or_1	vector	AR_3	vector	SSO
13	Sz_2	initial	Or_2	initial	AR_2	initial	SSO
14	Sz_3	$\lambda\,\Lambda(A_{ref})$	Or_1	$RV(A_{ref})$	AR_2	$\Delta(A_{ref})\,D$	SSO

The distribution of Aspect-ratio values over the initial and the optimal mesh may be either isotropic (equal to unity) or anisotropic (variable). Except for some special cases, most meshes do not have Aspect-ratio distributions that are exactly isotropic. However, if the Aspect-ratios are nearly unity (i.e., the distribution of aspect ratios has a small standard deviation from 1), we shall call the Aspect-ratio distribution isotropic. The three uses of the Aspect Ratio factor are:

AR_1. Create isotropic Aspect-ratio distribution ($\Delta_{new} = I$).

AR_2. Preserve the Aspect-ratio distribution of the initial mesh (Δ_{init}),

AR_3. Create a particular anisotropic Aspect-ratio distribution (Δ_{new}).

One could also have uses in which one is indifferent to the aspect ratio factor, but this is not included since none of our quality metrics is invariant to aspect ratio.

One can see from these use-lists that there are $4 \times 3 \times 3 \times 3 = 108$ combinations of uses of the four factors. An explicit enumeration of the combinations isn't particularly illuminating, especially since many of them do not seem likely to occur in practice. Instead, we focus on the known applications of mesh optimization, and relate them to the uses listed above in order to illustrate that TMP, with its factorization method of creating Target-matrices has the potential to address these applications. Table 1, along with the explanation below, summarizes the map between the applications and the TMP matrix factors and local metrics. Also shown is the likely source of the application data needed to construct the factors. For most of the applications, the Skew-matrix use is Sk_1 because in nearly every situation, the skew corresponding to the ideal element type is either explicitly desired or at least makes a good default in the absence of more detailed in formation. As a result, the Table does not include a Skew column.

An explanation of the Table follows:

1. **Shape Improvement**.
   Create isotropic elements whose Shape corresponds to the ideal isotropic element. The initial mesh has some unacceptable Skew. The initial Aspect Ratios are not extreme and the Size and Orientation is acceptable. The optimal mesh has less Skew and the Aspect Ratios are more isotropic, while the initial Size and Orientation are more or less preserved. Useful on most isotropic meshes.

2. **Skew Improvement**.
   Create elements whose Skew corresponds to the ideal element. The initial mesh has some unacceptable Skew and the Aspect Ratios, anisotropic or not, are acceptable. The Size and Orientation in the initial mesh is acceptable. The optimal mesh has less

Skew, while Aspect Ratio, Size, and Orientation are more or less preserved. Useful on most meshes, but especially if they are anisotropic.

3. **Mesh Morphing - I**.
Create a mesh whose local Shapes are similar to those of the reference mesh. This Target and metric are appropriate if the global deformation involves significant changes in domain Size (unknown), Orientation (unknown), and/or Aspect Ratio (known). Local aspect ratios in the optimal mesh will be roughly the same as those of the reference mesh when D, a known diagonal stretching of the domain, is the identity matrix. If D is not the identity, then the local aspect ratios will roughly be those of the reference mesh times the stretching factor. This Target is equivalent to $W_k = (A_{ref})_k\, D$ (see subsection 4.6).

4. **Shape Improvement with Size-Equidistribution**.
Create equal-Sized elements whose Shape corresponds to the ideal isotropic element. The initial mesh is not satisfactorily homogeneous and may contain unacceptable Skew or mild anisotropy. The initial Orientation is acceptable. The optimal mesh is more Size-homogeneous, with less Skew and anisotropy. Used to improve the Shape of isotropic meshes and equidistribute the Size. This case includes the minimum edge length example in the previous section.

5. **Anisotropy-Preserving Skew Improvement & Size-Equidistribution**.
The initial mesh has satisfactory Orientation and Anisotropy, but lacks sufficient Size-homogeneity, as well as perhaps having local Skew defects. The optimal mesh is more Size-homogeneous, has less Skew, and the initial mesh anisotropy is roughly preserved. Useful on anisotropic meshes that are not Size-adapted (e.g., a 3D swept mesh).

6. **Shape Improvement of Size-Heterogeneous Meshes**.
Create elements whose Shape corresponds to the ideal isotropic element, while preserving the initial (heterogeneous) Size-distribution. The initial mesh has a heterogeneous Size distribution which must be preserved, but whose Shapes need improvement toward the ideal isotropic element; initial Orientation is acceptable. The optimal mesh has the same heterogeneous Size distribution as the initial mesh, but has less skew and the local aspect ratios are closer to unity. Used in shape improvement of isotropic Size-adapted meshes.

7. **Skew Improvement for Heterogeneous, Anisotropic Meshes**.
The initial mesh has acceptable Size-heterogeneity, local anisotropy, and random Orientation; the mesh contains significant local Skew. The optimal mesh has less Skew, with Size and Aspect Ratios preserved. Similar to Application 2. except that Size is explicitly preserved. Useful on Skewed initial meshes whose Size and Aspect ratios are pre-adapted (e.g., a biased, structured mesh).

8. **Surface Mesh Size-Adaptivity to Curvature**.
The initial surface mesh is roughly homogeneous and isotropic. The optimal mesh has a local Size which is proportional to the (absolute value of the) local surface curvature, and has ideal isotropic Shape.

9. **Size-based r-Adaptivity**.
The initial mesh is roughly homogeneous and isotropic. The optimal mesh has a local Size-distribution which is proportional to the magnitude of a local error estimate or indicator, and has ideal isotropic Shape.

10. **Mesh Morphing - II**.
Same as Application 3 except that the domain deformation includes a known uniform scaling factor λ. The Target is $W_k = \lambda\, (A_{ref})_k\, D$, as described in subsection 4.6.

11. **Tensor-based r-Adaptivity**.
The initial mesh has a favorable topology so that the adaptive procedure will be effective. The optimal mesh has a Size-distribution based on the determinant of a

symmetric, positive definite tensor, an Orientation based on the tensor eigenvectors, and a Shape based on the ratio of the eigenvalues. Can be used in solution-adaptivity or geometric-adaptivity.

12. **Mesh Alignment**.
The initial mesh is roughly homogeneous and isotropic, with unsatisfactory Orientation. The optimal mesh has Orientation and Aspect Ratio based on a given vector field, has little Skew, and a constant Size-distribution. Boundary nodes should generally be allowed to move in order for this to work well. Used in applications such as fusion and electromagnetics whose dependent variables are vector fields.

13. **Skew-improvement of Aligned Mesh**.
The initial mesh has acceptable Size, Orientation, and Aspect Ratio, but is unacceptably Skewed. The optimal mesh is the same, with reduced Skew. Perhaps useful in ALE rezone calculations or other applications where the initial mesh is aligned and highly adapted to the solution.

14. **Mesh Morphing - III**.
Same as Application 10 except that the domain deformation includes a known constant rotation factor R. The Target is equivalent to $W_k = \lambda\, R\,(A_{ref})_k\, D$, as described in subsection 4.6.

For applications using the Sh or SS metrics, one can use $V = I$ instead of V_{init}, if this is more convenient, because the metrics are invariant to Orientation.

The examples cited in the Table only illustrate the potential of the Target-matrix paradigm to address a wide range of applications using only a few metrics, along with an automatic Target construction scheme. Essentially, we've shown that the Target-matrix paradigm is mainly about the creation of custom-built smoothers (optimizers) to address specific applications and which, being based on the same mathematical framework, permit rapid software implementation. Application 1 has been already developed within the Mesquite code [13] and has been a success. Exploratory papers have been published on applications 3 and 12. Additional work on the fourteen applications in the form of numerical optimization of application meshes remains in order to fully develop and illustrate their potential (this will be the subject of future papers). We do not claim that the Table is an exhaustive enumeration of all the potential applications of mesh optimization, but we believe that it may cover a large part of the application space.

There are two situations in mesh optimization which are handled through the choice of local metric instead of through the Target, namely, untangle an initially tangled mesh, or, ensure that an initially untangled mesh remains untangled during optimization. In the first case, we must use non-barrier metrics, while the second case calls for a barrier metric. This comment applies to each of the applications in the Table.

The applications mentioned in the Table share a common feature, namely, that each sample point is emphasized equally. This means that in the optimal mesh, the quality (as encoded in the Target-matrix and local metric) is equidistributed. A future paper will discuss how to vary the emphasis from one sample point to the next so that quality can be controlled most in regions of the domain where it is most important.

References

[1] P. Knupp, *Formulation of a Target-Matrix Paradigm for Mesh Optimization*, SAND2006-2730J, Sandia National Laboratories, 2006.

[2] P. Knupp, *Local 2D Metrics for Mesh Optimization in the Target-matrix Paradigm*, SAND2006-7382J, Sandia National Laboratories, 2006.

[3] P. Knupp and E. van der Zee, *Convexity of Mesh Optimization Metrics Using a Target-matrix Paradigm*, SAND2006-4975J, Sandia National Laboratories, 2006.

[4] P. Knupp, *Analysis of 2D Rotational-invariant Non-barrier Metrics in the Target-Matrix Paradigm*, SAND2008-8219P, Sandia National Laboratories, 2008.

[5] P. Knupp, *Label-invariant Mesh Quality Metrics*, to be published in the Proceedings of the 18th International Meshing Roundtable, Springer, 2009.

[6] L. Freitag and P. Knupp, *Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number*, 8th International Meshing Roundtable, pp247-258, 1999.

[7] P. Knupp, *Updating Meshes on Deforming Domains*, Communications in Numerical Methods in Engineering, 24:467-476, 2008.

[8] P. Knupp, *An analysis of binary trade-off metrics in the Target-matrix paradigm*, manuscript, 2009.

[9] P. Knupp, *Algebraic mesh quality metrics*, SIAM J. Sci. Comput., Vol. 23, No. 1, pp193-218, 2001.

[10] P. Knupp, *Jacobian-weighted Elliptic Grid Generation*, SIAM J. Sci. Comput., Vol. 17, No. 6, 1996.

[11] P. Knupp, *Mesh Generation Using Vector Fields*, J. Comp. Phys., 119, pp142-148, 1995.

[12] P. Knupp, L. Margolin, and M. Shashkov, *Reference Jaocobian Optimization-based Rezone Strategies for Arbitrary Lagrange-Eulerian Methods*, J. Comp. Phys., pp93-128, Vol. 176, No. 1, 2002.

[13] M. Brewer, L. Diachin, P. Knupp, and D. Melander, *The Mesquite Mesh Quality Improvement Toolkit*, p239-250, Proceedings of the 12th International Meshing Roundtable, Santa Fe NM, 2003.

Figure 4: Initial (Top) and Optimized (Bottom) Paved Mesh